

Resum

Aquest Treball de Fi de Grau té com a objectiu l'estudi i l'aplicació de les funcionalitats que ens ofereix la recentment coneguda tecnologia de l'Internet de les Coses.

El nou corrent d'innovacions de que està gaudint i gaudirà la tecnologia contemporània gràcies a aquest fenomen ha estat estudiat mitjançant un mòdul per a desenvolupadors TD1204 de l'empresa Telecom Design. En aquest treball s'explica com és aquest mòdul, de quina manera s'ha d'operar amb ell i què s'ha de fer per expressar les seves capacitats i transportar-les a qualsevol lloc web.

En aquest cas, el camp d'aplicació d'aquesta nova tecnologia han estat els vehicles elèctrics. Agafant com a referència l'Smart Moto Challenge de l'ETSEIB, s'han dissenyat funcions per controlar aspectes del vehicle tals com la bateria i la posició des de qualsevol aparell amb connexió a internet.

Donada la poca informació que hi ha en aquests moments sobre l'Internet de les Coses, ha estat molt complicat trobar la documentació necessària per implementar un codi que realitzés les accions adients per rebre els missatges de la placa de desenvolupament TD1204. Han fet falta moltes hores d'estudi i moltes proves per acabar dissenyant una pàgina web que mostri per pantalla les dades que s'envien des de la placa a un servidor.

Amb el motiu que acabo d'esmentar, aquest treball vol ser un ajut per pròxims projectes relacionats amb l'àmbit de la connectivitat, que trobaran aquí uns passos a seguir i unes funcions preparades per dissenyar aplicacions que ens facilitin i automatitzin moltes tasques del dia a dia en un futur no molt llunyà.



Sumari

RESUM	1
SUMARI	3
1. GLOSSARI	5
2. PREFACI	7
2.1. Origen del projecte	7
2.2. Motivació	7
2.3. Requeriments previs	8
3. INTRODUCCIÓ	9
3.1. Objectius del projecte.....	10
3.2. Abast del projecte	10
4. HARDWARE	11
4.1. TD1204. Dispositiu i EVB (placa d'avaluació)	11
4.2. Perifèrics	13
4.2.1. Acceleròmetre.....	13
4.2.2. Sensor de temperatura	14
4.2.3. GPS	14
4.2.4. Antena de RF.....	17
5. ACCÉS A LES DADES	20
5.1. Funcionament de les APIs	21
5.2. Funcions cURL.....	22
5.3. Internet of Things API	24
5.3.1. Device API	24
5.3.2. Developer API.....	26
5.4. Implementació en format PHP	26
6. PÀGINA WEB	34
6.1. Nivell de bateria.....	34
6.2. Localització GPS	37
6.3. Quadre de text i qüestionari	39
6.4. Format i distribució	41
CONCLUSIONS	42

AGRAÏMENTS	43
-------------------	-----------

BIBLIOGRAFIA	44
---------------------	-----------



1. Glossari

IoT: "Internet of Things"

API: "Application Programming Interface"

HTML: "HyperText Markup Language"

PHP: "Hypertext Preprocessor"

GPS: "Global Positioning System"

TDD: "Test-Driven Development"

RF: "Radio Frequency"

LVTTL: "Low Voltage Transistor-Transistor Logic"

UART: "Universal Asynchronous Receiver-Transmitter"

PWM: "Pulse-Widht Modulation"

GPIO: "General Purpose Input/Output"

AT commands: "Attention commands"

GNSS: "Global Navigation Satellite System"

GLONASS: "Global Orbiting Navigation Satellite System"

SBAS: "Satellite-Based Augmentation System"

HDOP: "Horizontal Diluition Of Precision"

FSK: "Frequency Shift Keying"

GMSK: "Gaussian Minimum Shift Keying"

MSB: "Most Significant Bit"

LSB: "Least Significant Bit"

URL: "Uniform Resource Locator"

CURL: “Client URL Request Library”

HTTP: “Hypertext Transfer Protocol”

FTP: “File Transfer Protocol”

HTTPS: “Hypertext Transfer Protocol Secure”

JSON: “JavaScript Object Notation”

XML: “eXtensible Markup Language”

2. Prefaci

2.1. Origen del projecte

Des de fa poc temps, ha anat creixent una nova corrent de desenvolupament tecnològic gràcies al IoT o “Internet of Things”. Aquest concepte es refereix a la interconnexió entre objectes quotidians i internet mitjançant una comunicació de baix consum, permetent a qualsevol aparell enviar i rebre informació de la xarxa.

Són innumerables les aplicacions que són possibles gràcies a aquesta tecnologia, des d'alarmes d'incendis mòbils connectades amb les estacions de bombers, fins a frigorífics que compren per internet els productes que ja hem consumit.

Al mateix temps, el vehicle elèctric esdevé cada dia més necessari per tal de reduir la contaminació a les grans ciutats i proporcionar una major eficiència energètica respecte als vehicles convencionals.

Aquest projecte té relació amb aquests dos grans canvis de la tecnologia contemporània. Vol connectar els vehicles elèctrics amb internet mitjançant aplicacions que facilitin als usuaris l'ús del seu automòbil.

2.2. Motivació

A l'hora d'organitzar un viatge amb un vehicle propi, és convenient assegurar-se de tenir autonomia suficient per realitzar-lo, o saber on pots aturar-te a reomplir el dipòsit si no és el cas. Amb un vehicle elèctric aquesta mesura té més importància donat que avui en dia hi ha menys punts de recàrrega i l'autonomia d'aquests automòbils és menor.

La motivació d'aquest projecte, per tant, és la de brindar una eina als usuaris dels vehicles elèctrics per tal de que puguin saber en tot moment quin és el nivell de bateria del que disposen i a la vegada poder visualitzar un seguiment GPS del seu cotxe o moto, per tal de facilitar l'ús i augmentar la seguretat del mateix.

A més a més, una altra gran motivació és la d'obrir les portes a l'Internet de les Coses amb aquest projecte i així ajudar a desenvolupar l'inevitable corrent d'innovacions que arribaran en un futur molt proper.

2.3. Requeriments previs

Abans de posar-se a pensar en possibles aplicacions ha estat necessari familiaritzar-se amb les eines que es faran servir per a dissenyar-les i aplicar-les.

Per començar, s'han analitzat els diferents components i característiques de la placa de desenvolupament TD1204, s'ha instal·lat el software necessari (terminal “*Realterm*” i “u-center” de la companyia Ublox per a descodificar les trames enviades per l'antena GPS) i s'han estudiat les diferents comandes que es poden enviar a la placa.

També s'ha estudiat la documentació de la web de Telecom Design per conèixer les diferents APIs que podrem utilitzar i com implementar-les en un lloc web. I s'han seguit tutorials on-line pel disseny de pàgines web mitjançant llenguatge HTML i PHP.

Per últim, ha estat molt interessant informar-se d'un treball de fi de màster i un de fi de grau que han realitzat tant abans com al mateix temps que aquest. Tant el treball de fi de màster: *Internet de les coses: una primera aproximació utilitzant la tecnologia SIGFOX* [1], realitzat per Israel Terán Lozano, com el treball de fi de grau: *Internet de les coses aplicat a la millora del servei de Bicing de Barcelona* [2], d'Albert Torres Pons, han estat claus per poder realitzar aquest projecte.

3. Introducció

A Espanya, segons la DGT, el parc d'automòbils superava ja els 30 milions d'unitats el 2014, amb una mitjana d'edat de 11,3 anys [3]. No fa falta ser alarmista per pensar que tots aquests vehicles aporten una gran contribució en la contaminació de les grans ciutats i per tant agreugen el ja famós escalfament del nostre planeta. Pel que fa al preu dels combustibles fòssils, en tant sols 11 anys (de Gener de 2002 a Gener de 2013), el preu del barril de petroli Brent ha passat de 19,42 a 112 dòlars, fent augmentar consegüentment el preu de la benzina que consumim [4].

És per això que el vehicle elèctric està obtenint tanta importància en aquests últims anys. Per una banda guanyem ecològicament aprofitant que part de l'energia elèctrica que consumim ve d'energies renovables i netes, i per altra banda, econòmicament tant per les ajudes que l'estat subministra per impulsar el vehicle elèctric, com pel menor cost de moure's propulsat per electricitat, que en el cas dels utilitaris, no supera els 2 euros cada 100 kilòmetres [5].

No obstant, al tractar-se d'una tecnologia relativament nova, no pot competir encara amb els vehicles de combustió pel que fa a l'autonomia, no gaudeix del mateix nombre de infraestructures per realitzar les recàrregues, ni aquestes recàrregues són ni molt menys tan ràpides.

Aquest gran potencial de millora es podria aprofitar per fer aplicacions que ajudessin a l'usuari a tenir un control sobre el seu vehicle i els punts de recàrrega més propers. Malauradament, una connexió a internet des del cotxe per informar a l'usuari dels problemes que aquest pugui patir, requereix d'una tarifa mensual de 3G o 4G amb una gran companyia que ocasionaria un sobrecost innecessari, fins fa ben poc.

Amb l'arribada del Internet of Things, podem gaudir d'una connexió a internet per rebre i enviar un nombre limitat però suficient de dades per a infinitat d'aplicacions. En un futur no molt llunyà, gràcies a aquesta tecnologia conviurem amb una xarxa d'objectes quotidians interconnectats que ens faran que moltes de les nostres tasques siguin més fàcils i intuïtives.

Aquest treball pretén aproximar aquests dos sectors de la més nova tecnologia per donar solucions o alternatives a la indústria del vehicle elèctric mitjançant un prototip completament funcional que implementi l'Internet de les Coses.

3.1. Objectius del projecte

El principal objectiu d'aquest projecte és el de dissenyar aplicacions per facilitar l'ús dels vehicles elèctrics als seus usuaris mitjançant informació en tot moment de la posició i estat de la bateria de l'automòbil. En aquest cas basarem les aplicacions en la nova motocicleta de la Barcelona Smart Moto Challenge de l'ETSEIB, la BRUC'03.

Un altre propòsit que es vol dur a terme és l'aprofundiment en el món de l'IoT, una tecnologia en ple auge i que cada dia evoluciona i creix fent pensar que suposarà una revolució en el món de la tecnologia en els propers anys. Aquest projecte és una gran oportunitat per poder veure de ben a prop com aquesta tecnologia pot, tant afectar de manera positiva a molts dels serveis que s'ofereixen actualment, com obrir la porta per nous serveis que ens facin la vida més fàcil.

Per assolir aquests objectius utilitzarem la placa de desenvolupament TD1204, fabricada per l'empresa SIGFOX. Aquesta placa porta incorporat un microcontrolador i varis perifèrics, tals com un receptor GPS, un acceleròmetre i un transceptor de radiofreqüència.

S'utilitzarà un ordinador portàtil com a "host" per comandar la placa i diversos perifèrics addicionals de manera que es pugui transmetre tota la informació sobre l'estat de la motocicleta en temps real i es pugui actuar en conseqüència d'una manera ràpida i eficient.

3.2. Abast del projecte

El propòsit final d'aquest projecte és el d'estudiar les funcions disponibles per accedir remotament a les dades que envia la placa TD1204, els processos a seguir per utilitzar aquestes funcions en un lloc web i per tant, la manera com crear aquest lloc web per poder rebre a internet la informació de l'Smart Moto Challenge a temps real.

L'abast d'aquest projecte inclou el tractament dels missatges des de que són enviats per la placa fins que són mostrats per una pàgina web. La implementació del mòdul a l'Smart Moto, així com l'automatització dels missatges són conceptes que no estan inclosos en aquest projecte.

4. Hardware

4.1. TD1204. Dispositiu i EVB (placa d'avaluació)

El TD1204 és un dispositiu d'alt rendiment i baix corrent. El seu funcionament sense fils juntament amb l'ampli rang de tensions d'alimentació (2,3-3,6V) i al poc consum de corrent (2.5 μ A en mode idle) fa que sigui una bona opció alimentar-la mitjançant bateries, fet que li aporta una gran autonomia.

Opera com un transceptor dúplex per divisió de temps (TDD), transmetent i rebent dades alternadament.

El TD1204 de Telecom Design incorpora:

- Un processador ARM Cortex M3 de banda base de 32 bits.
- Un transceptor de RF.
- Un receptor GPS d'alta eficiència.
- Acceleròmetre 3D amb detecció de moviment i caiguda lliure.
- Interfícies digitals i analògiques.
- Un LVTTTL UART de baixa potència.
- Un bus I2C.
- Timers amb capacitats per contar pols (input) o PWM (output).
- Dos convertidors A/D d'alta resolució i rapidesa i un convertidor D/A.
- Varis pins d'entrada/sortida (GPIO).

El dispositiu porta integrat un intèrpret de comandes "AT" per facilitar la interconnexió amb un altre dispositiu que opera com a host enviant les comandes en mode AT i rebent les respostes a aquestes comandes. D'aquesta manera, podem enviar-li les comandes mitjançant un ordinador, un microprocessador extern, o simplement programant el ARM Cortex M3 intern per que aquest es comuniqués amb la placa.

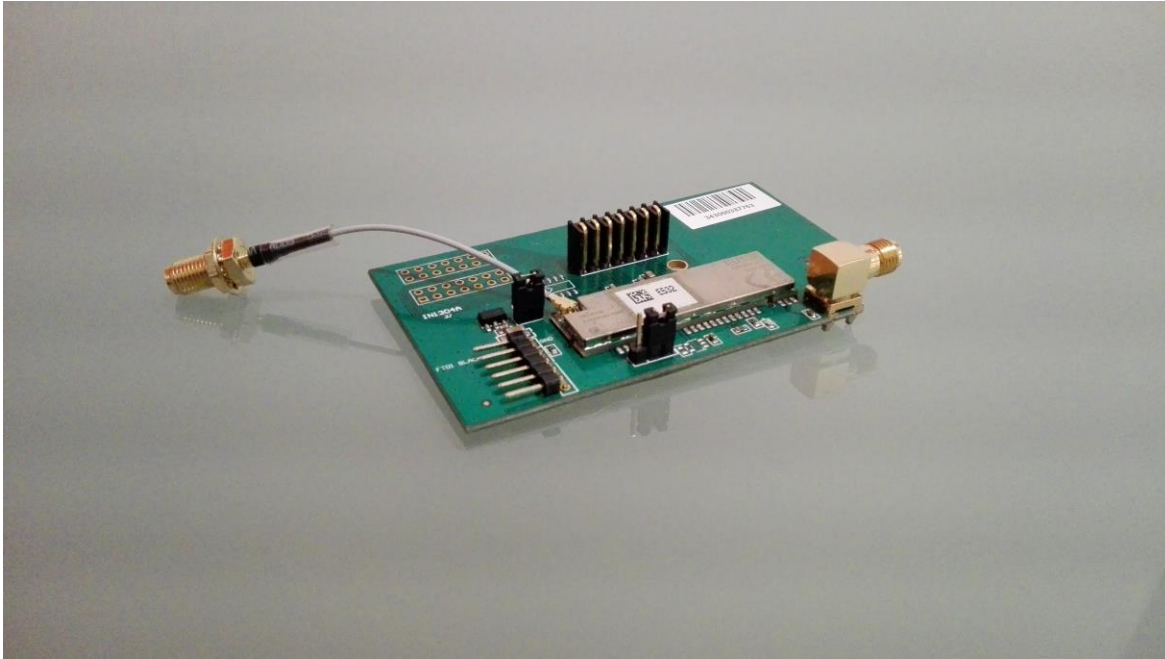


Fig. 4.1. Mòdul TD1204

Abans de començar a dissenyar les aplicacions s'ha utilitzat el terminal *RealTerm* i el manual de referència del dispositiu, per familiaritzar-se amb les comandes disponibles i preparar el dispositiu per a que la transmissió/recepció bidireccional de dades funcioni correctament [6]. Amb aquesta finalitat, s'han definit alguns paràmetres utilitzant les comandes:

AT → retorna "OK".

ATE_x → determina si les comandes AT es mostren en el terminal ($x=1$) o no ($x=0$), de manera que només es veuen les respostes. Aquest fet és especialment important, ja que a l'hora d'emmagatzemar les respostes en una cua, aquesta no s'omple innecessàriament amb les comandes que s'envien.

ATQ_x → determina si la placa envia codis de resultat ($x=0$) o no ($x=1$). És evident que es necessiten les respostes.

ATV_x → determina si la resposta utilitza salts de línia (<CL>) i salts de carro (<CR>) i està en mode verbós ($x=1$) o si no n'utilitza i està en mode numèric ($x=0$). Per les aplicacions és necessari introduir ATV1.

ATXx → determina si la resposta es transmet en mode normal ($x=0$) o en mode estès ($x=1$). S'utilitzarà el mode estès.

AT&W → guarda la configuració dels paràmetres.

Tan per rebre dades dels diferents perifèrics com per enviar missatges mitjançant l'antena de RF, s'utilitzaran diferents comandes que s'aniran explicant més endavant.

4.2. Perifèrics

4.2.1. Acceleròmetre

El mòdul TD1204 conté un acceleròmetre capaç de mostrar les acceleracions en un instant determinat en 3 direccions ortogonals. Utilitzant aquest perifèric podríem saber si la moto està en marxa o aturada, o fins i tot si ha patit un accident, degut a les grans acceleracions puntuals que experimentarà.

La comanda que s'ha d'enviar a la placa TD1204 per accedir a aquestes dades és:

ATS650 = <permetre>, <alimentació>, <cadència>, <escala>, <filtre>

On:

- <permetre> és 0 si el monitoratge està desactivat i 1 en cas contrari.
- <alimentació> és 0 si s'utilitza en mode d'alta resolució o 1 en mode de baixa resolució (consumint menys).
- <cadència> es refereix a la freqüència de presa de dades i pot anar de 1 a 4, sent els valors possibles 1, 10, 25 i 50 Hz respectivament.
- <escala> es refereix a l'escala de l'acceleròmetre i pot prendre els valors 2 ($\pm 2g$), 4 ($\pm 4g$), 8 ($\pm 8g$) i 16 ($\pm 16g$).
- <filtre> es refereix a l'activació (1) o no (0) d'un filtre passa-alt que suprimeix l'efecte de la gravetat en les lectures de l'acceleròmetre.

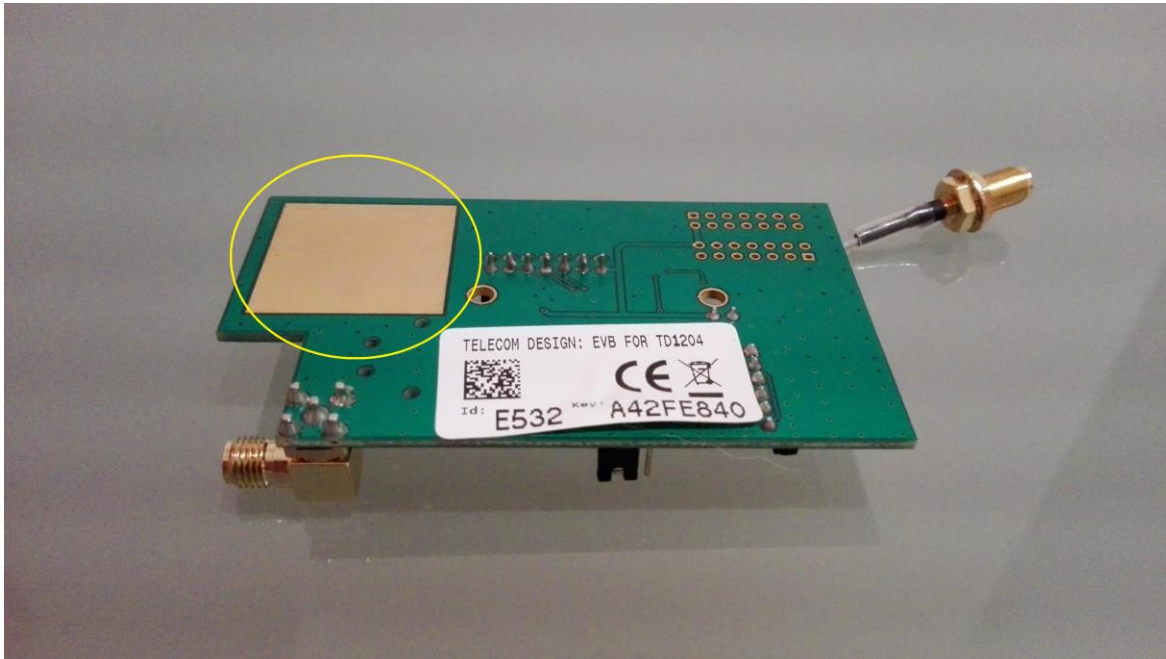


Fig. 4.2. Acceleròmetre incorporat a la TD1204

En aquest projecte, les funcions que es volien desenvolupar no necessitaven d'aquest acceleròmetre, per tant no l'utilitzarem. En un projecte futur podríem utilitzar-lo per programar avisos de possibles accidents o per saber quan el vehicle esta parat.

4.2.2. Sensor de temperatura

Es tracta d'un termòmetre digital que s'utilitza per alertar de temperatures límit que puguin malmetre el dispositiu. Es pot programar per configurar aquestes temperatures límit, tant per d'alt com per baix, i així rebre un avís quan la temperatura del mòdul sobrepassi aquests llindars.

També podem conèixer la temperatura del mòdul utilitzant aquesta comanda:

- AT126 ➔ retorna el mòdul de la temperatura en °C.

Tot i ser un perifèric molt interessant, tampoc utilitzarem aquesta funció per el nostre projecte.

4.2.3. GPS

La antena GPS és un dels perifèrics més interessants per aquest projecte. Amb ella podrem saber la posició de la nostra motocicleta en tot moment, cosa que afegirà seguretat a l'usuari, que en cas de pèrdua o robatori del seu vehicle només haurà de consultar la

nostra pàgina web per conèixer la seva posició.

Les principals característiques d'aquesta antena són:

- Suport múltiple GNSS:
 - GPS/GLONASS.
 - Augment de serveis SBAS.
- Consum d'energia:
 - 22 mA en mode adquisició.
 - 15 μ A en mode reserva (backup).
- Sensibilitat:
 - 56-motor de canal.
 - 162 dBm en mode rastrejador.
 - -148 dBm per a arrancada en fred.

La comanda que s'ha d'enviar a la placa TD1204 és:

AT\$GPS=<mode>, <nº mínim satèl·lits>, <hdop màxim>, <temps límit>, <mode final>, <NMEA>

On:

- <mode> pot ser 0 si s'apaga, 1 si s'encén o 2 si es deixa en mode "*hardware backup*".
- <nº mínim satèl·lits> és el nombre mínim de satèl·lits que es volen utilitzar per a calcular la posició. Pot anar de 0 a 255.
- <hdop màxim> és el màxim valor del paràmetre hdop per a detectar una posició com a vàlida. Va de 0 a 65535, sent aquest el valor de hdop multiplicat per 100. Si es posa a 0 mai es trobarà una posició vàlida.
- <temps límit> Valor límit després del qual, encara que no s'hagi trobat una solució adequada, es mostrarà la posició més pròxima que s'aconsegueixi. Compren un rang de valors entre 0 i 65535 segons, significant aquest últim valor que no s'utilitza

un temps límit.

- <mode final> Indica el mode en el que es deixarà el GPS després d'haver transmès la posició. 0 si s'apaga, 1 si es deixa encès o 2 si es deixa en mode *"hardware backup"*.
- <NMEA> Indica la informació que es vol llegir del GPS. 0 si no es vol cap informació, 1 si es volen tots els missatges en codi NMEA o 2 si només es volen tan sols els del tipus GPGGA, sent aquesta la frase que conté la informació a utilitzar.



Fig. 4.3. Antena externa de GPS

Per poder utilitzar el GPS, aquest s'encendrà en el moment que s'encengui el dispositiu, s'esperarà uns dos minuts per que aquest hagi establert connexió amb els satèl·lits i proporcioni una posició vàlida, i aquesta posició serà enviada pel mòdul a internet.

Així, just encendre el programa, s'enviarà la comanda:

AT\$GPS=1,16,900,360,0,1

I quan es vulgui enviar la posició GPS, s'utilitzarà la comanda:

AT\$GSND

Aquesta comanda agafa la última posició GPS vàlida i la envia al servidor de Sigfox. Aquest

missatge contindrà dades com la latitud, la longitud i la altitud.

Cal comentar que s'han tingut alguns problemes amb l'antena. El primer és la lentitud a l'hora de connectar amb els satèl·lits. Normalment s'ha d'esperar uns dos minuts per aconseguir rebre dades vàlides. A més, si es connecta el dispositiu en espais tancats és probable que el GPS no arribi mai a donar dades vàlides.

4.2.4. Antena de RF

Sense aquest perifèric, res hauria estat possible, ja que l'antena de RF és l'encarregada d'enviar tots els missatges que necessitem al Device Dashboard de Telecom, plataforma del servidor de dades situat a França. Allà podrem consultar-los directament, o utilitzar les APIs que ens ofereixen per redreçar els missatges a una pàgina web.

Les principals característiques d'aquesta antena són:

- Rang de freqüència: ISM 868 MHz.
- Sensibilitat de recepció: -126 dBm.
- Modulació:
 - (G)FSK, 4(G)FSK, GMSK.
 - OOK.
- Potència de sortida màxima: +14 dBm.
- Consum de potència estant activa:
 - 22 μ A RX (mode finestra).
 - 37 mA TX @ +10 dBm.

Aquesta antena la utilitzarem tal i com hem vist per enviar la posició GPS de la motocicleta, però també per enviar un missatge codificat que contingui l'estat de la bateria d'aquesta. En aquest projecte, per falta de temps i de disposició de la BRUC'03, no s'ha acoblat la placa a la bateria per poder consultar-ne l'estat, si no que com a primer pas, s'ha decidit enviar manualment missatges aleatoris amb percentatges d'aquesta bateria per estudiar-ne el seu tractament.



Fig. 4.4. Antena externa de RF

Com encara no s'ha decidit si el missatge enviarà directament el valor digital que rebí del convertidor analògic-digital del mòdul TD1204, o tractarà aquest valor per enviar ja el valor percentual de la bateria, s'han dissenyat dos tipus de missatge per simular aquests dos casos.

En ambdós missatges, la comanda per enviar-los serà:

AT\$RAW= AB CD EF GH

On cada parella de lletres serà una parella de dígit hexadecimals que una vegada enviats, seran transformats també a valors ASCII.

Pel cas en que enviem el valor percentual, construirem el missatge per tal que indiqui que conté el valor en ASCII i seguidament mostri tres caràcters del 0 al 9, un per les centenes, un per les desenes i l'últim per les unitats. Com a exemple, si volem enviar un missatge on indiqui que la bateria és del 50%, el text hauria de ser: SoCASC050. Però abans d'enviar la comanda, hem de passar aquests valors a hexadecimal, pel que la comanda final seria:

AT\$RAW= 53 6f 43 41 53 43 30 35 30

Pel que fa a l'altre cas, la construcció serà una mica més complicada de fer i d'entendre. Començarem el missatge amb un text similar al d'abans, SoCDIG, que en aquest cas

indica que els valors seran digitals, i després enviarem dos parelles de caràcters hexadecimals, als primers els hi direm X i Y, mentre que als segons els hi direm K i L.

Per una banda, X (MSB) i Y(LSB) formaran un número proporcional al SoC de la bateria que tingui la motocicleta seguint la fórmula: $256 \cdot X + Y$. Per l'altra banda, K i L formaran un altre número depenent del nombre de bits del convertidor A/D, seguint una fórmula semblant que mostrarà el fons d'escala i per tant el nombre màxim de bits que poden formar X i Y.

A l'hora de calcular el tant per cent de bateria que resta (T), haurem de seguir aquesta fórmula:

$$\text{SoC} = 100 \cdot (256 \cdot X + Y) / (256 \cdot K + L) = T \%$$

D'aquesta manera, per enviar un missatge de tipus digital, primer passarem la paraula digital del convertidor A/D i el fons d'escala a hexadecimal. Per exemple, si utilitzem un convertidor de 10 bits, el fons d'escala serà de $2^{10} - 1 = 1023$, i si volem indicar que el tant per cent de bateria és del 50%, $256 \cdot X + Y = 512$. Per tant, passat ja a hexadecimal, $X Y = 02 00$ i $K L = 03 FF$.

Després de passar també el text SoCDIG a hexadecimal, aquesta seria la comanda per enviar el missatge al servidor de Sigfox:

AT\$RAW= 53 6f 43 44 49 47 02 00 03 FF

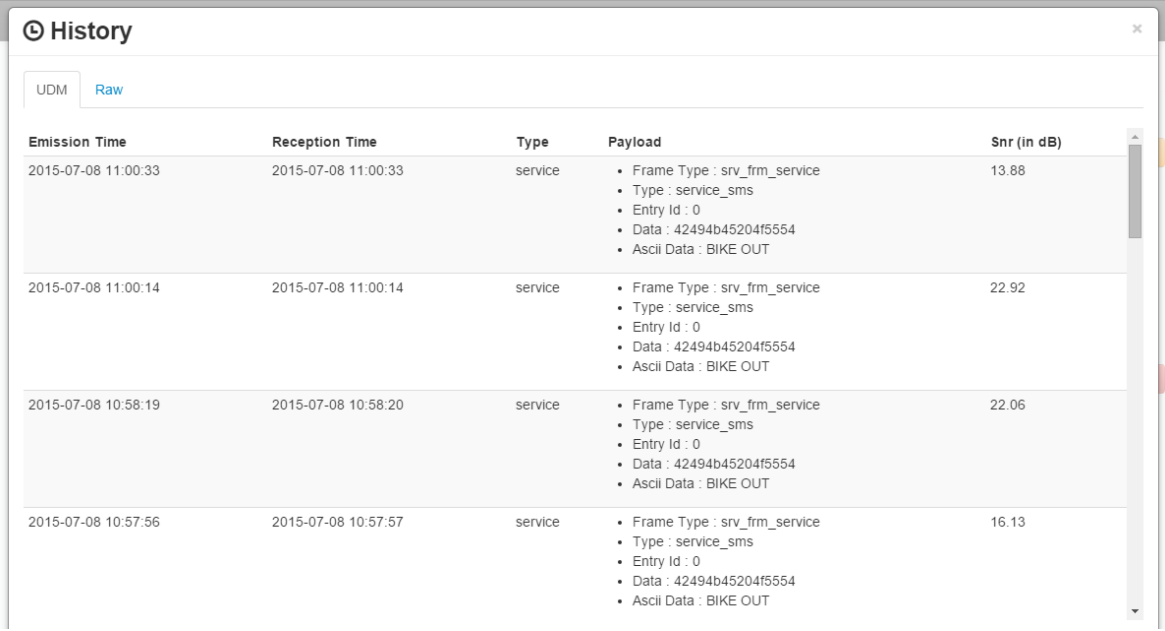
En un futur, un microprocessador exterior o el mateix microcontrolador de la TD1204 estarà programat per realitzar cada 30 minuts, per exemple, les 3 comandes que hem vist que hem d'enviar. Primer la necessària per encendre el GPS, dos minuts després la que envia l'estat de la bateria i finalment la que envia la posició de la placa.

No obstant, per fer funcionar el projecte en aquests moments, enviarem les comandes des d'un ordinador portàtil mitjançant el programa *RealTerm* des de diferents ubicacions i amb diferents missatges pel que fa a la bateria.

5. Accés a les dades

Gràcies a un microcontrolador (intern o extern) en un futur, la placa enviarà dos missatges cada 30 minuts. El primer missatge contindrà el tant per cent de bateria de que disposa la motocicleta (SoC) i el segon, la ubicació d'aquesta.

Aquests dos missatges són enviats al servidor de Sigfox, situat a França. Per visualitzar aquestes dades, podem fer ús del Device Dashboard de Telecom [7], on després d'introduir l'identificador (ID) i la paraula de pas (Key) del mòdul, entrem a una pàgina web on podem revisar l'historial de missatges enviats per la placa, així com la data i hora de cada missatge. Fent servir aquest portal, podem fer proves per comprovar que els missatges són enviats correctament, però una vegada hem aconseguit aquest pas, necessitem una altra eina per poder llegir els missatges des de una pàgina web pròpia o des de una aplicació d'Android.



Emission Time	Reception Time	Type	Payload	Snr (in dB)
2015-07-08 11:00:33	2015-07-08 11:00:33	service	<ul style="list-style-type: none"> Frame Type : srv_frm_service Type : service_sms Entry Id : 0 Data : 42494b45204f5554 Ascii Data : BIKE OUT 	13.88
2015-07-08 11:00:14	2015-07-08 11:00:14	service	<ul style="list-style-type: none"> Frame Type : srv_frm_service Type : service_sms Entry Id : 0 Data : 42494b45204f5554 Ascii Data : BIKE OUT 	22.92
2015-07-08 10:58:19	2015-07-08 10:58:20	service	<ul style="list-style-type: none"> Frame Type : srv_frm_service Type : service_sms Entry Id : 0 Data : 42494b45204f5554 Ascii Data : BIKE OUT 	22.06
2015-07-08 10:57:56	2015-07-08 10:57:57	service	<ul style="list-style-type: none"> Frame Type : srv_frm_service Type : service_sms Entry Id : 0 Data : 42494b45204f5554 Ascii Data : BIKE OUT 	16.13

Fig. 5.2. Historial de missatges del mòdul vist des del dashboard de Telecom Design.

També a la web de Telecom podem llegir documentació de totes les APIs que podem utilitzar, tant per tenir accés al mòdul des de el nostre lloc web, com per manipular els missatges que està enviant la placa [8].

Des del meu punt de vista, aquesta documentació es molt limitada i no gaire clara. Per cada funció, hi ha una descripció breu, el mètode que s'ha d'utilitzar per fer-la funcionar

(Get, Post, Delete, etc), la URL a la que s'ha de dirigir aquest mètode, els paràmetres que fan falta adjuntar i finalment un exemple de resposta per saber en quin format és retornada, així com els errors que poden sorgir.

GET /iot/devices/crc.json

Description	Device Authentication operation				
Url	<code>https://sensor.insgroup.fr/iot/devices/crc.json</code>				
Method	GET				
Parameters	<ol style="list-style-type: none"> 1. sn <i>required</i> The module serial number 2. key <i>required</i> Key information appears either next to module sn information on product packaging sticker (8 hexa chars, case insensitive), either as a PAC sn given by your module provider (16 hexa chars, case insensitive). 				
Returns	Device token for use with corresponding header token Sample JSON response <pre>QWixhZGRpbjpvGvUcGVuIHNLc2FtZQ==</pre>				
	Return value definitions <table border="1"> <thead> <tr> <th>Field</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>token</td> <td>Base64 encoded token, to store and use for next individual remote Device WS resources</td> </tr> </tbody> </table>	Field	Description	token	Base64 encoded token, to store and use for next individual remote Device WS resources
Field	Description				
token	Base64 encoded token, to store and use for next individual remote Device WS resources				
Errors	404 - The serial number was not found 401 - The authentication information are missing or invalid				

Fig. 5.2. Documentació de l'API per obtenir el Token d'un mòdul

Com es pot observar a la Fig 5.2, en cap moment comenta en quin llenguatge s'ha de realitzar el mètode Get, ni esmenta un exemple de com es cridaria aquesta funció. La informació que podíem obtenir de la web era, per tant, molt escassa. Juntament amb el meu desconeixement d'aquests mètodes i al fet que per ser una tecnologia tant nova, buscar informació relacionada amb aquest tema per internet es feia molt difícil, el projecte va estar aturat en aquest punt durant molt de temps.

Després de rellegir la documentació moltes vegades, trobar exemples similars per la xarxa i sobretot fer moltes proves, però, es va trobar la manera d'utilitzar aquestes funcions.

5.1. Funcionament de les APIs

Es possible que molta gent, com jo abans de començar aquest projecte, hagi sentit alguna vegada parlar de les APIs sense saber realment que significa. La abreviatura API ve de interfície de programació d'aplicacions (Application Programming Interfaces), i dona nom a

la especificació formal sobre la manera en que un mòdul de un software es comunica o interactua amb un altre.

Dit d'una altra manera, les APIs són un conjunt de funcions, comandes i protocols informàtics, que permeten als desenvolupadors crear programes específics simplificant en gran mesura el treball necessari per realitzar-los. Aquestes, per tant, permeten utilitzar funcions predefinides per interactuar amb el sistema operatiu o amb un altre programa.

Per fer-les servir, per exemple en una pàgina web, va ser necessari estudiar el llenguatge PHP per saber com s'havien de programar les APIs que ens farien falta. Per simplificar el treball, es va fer servir una eina software per la transferència d'arxius, amb el nom de cURL.

5.2. Funcions cURL

CURL és una llibreria de funcions que serveixen per connectar amb servidors i així treballar amb ells. Com es pot deduir pel seu nom, aquesta llibreria treballa amb format URL i per tant, suporta els protocols més comuns com http, ftp, https entre d'altres.

Com va passar repetides vegades en aquest projecte, amb cURL descobria coses de les que no havia sentit parlar mai, havia de començar des de 0. Inicialment es van estudiar exemples i explicacions, més endavant es van provar algunes funcions bàsiques mitjançant una Shell de Python i finalment es van poder utilitzar les funcions necessàries a través de PHP per poder-les fer servir a la web.

En el nostre cas, cURL ens ajudarà a simplificar la crida de les APIs que necessitem per rebre la informació que envia la placa al servidor de Telecom Design. Primer però, anem a veure un llistat de les funcions que podem utilitzar de la llibreria cURL [9], explicarem breument quina és la seva funció i ja entrarem en detall amb aquelles que utilitzem en el codi de la nostra web més endavant:

- *curl_close* : Tanca la sessió cURL.
- *curl_copy_handle* : Copia el recurs cURL juntament amb totes les seves preferències.
- *curl_errno* : Retorna l'últim número d'error.
- *curl_error* : Retorna una cadena que conté l'últim error de la sessió.
- *curl_escape* : Funció URL que codifica l'*string* donat.
- *curl_exec* : Estableix una sessió cURL.

- *curl_file_create* : Crea un objecte *CURLFile*.
- *curl_getinfo* : Obté informació relativa a una transferència específica.
- *curl_init* : Inicia sessió cURL.
- *curl_multi_add_handle* : Afegeix un recurs cURL a un grup de recursos cURL.
- *curl_multi_close* : Tanca un grup de recursos cURL.
- *curl_multi_exec* : Executa les subconnexions del recurs cURL actual.
- *curl_multi_getcontent* : Retorna el contingut del recurs cURL si *CURLOPT_RETURNTRANSFER* està activat.
- *curl_multi_info_read* : Obté informació de les transferències en curs.
- *curl_multi_init* : Retorna un nou multi recurs cURL.
- *curl_multi_remove_handle* : Elimina un multi recurs d'un conjunt de recursos cURL.
- *curl_multi_select* : Espera activitat en qualsevol connexió en *curl_multi*.
- *curl_multi_setopt* : Configura una opció pel multi gestor cURL.
- *curl_multi_strerror* : Retorna un *string* amb la descripció del codi de l'error.
- *curl_pause* : Atura i reprèn una connexió.
- *curl_reset* : Reinicialitza totes les opcions d'un gestor de sessió *libcurl*.
- *curl_setopt_array* : Configura múltiples opcions per a una transferència cURL.
- *curl_setopt* : Configura una opció per a una transferència cURL.
- *curl_share_close* : Tanca un gestor cURL compartit.
- *curl_share_init* : Inicialitza un gestor cURL compartit.
- *curl_share_setopt* : Configura una opció per a un gestor cURL compartit.
- *curl_strerror* : Retorna un *string* que descriu el codi d'error donat.
- *curl_unescape* : Descodifica un *string* codificat de URL.
- *curl_version* : Obté la informació de la versió cURL.

Algunes d'aquestes funcions ens permetran inicialitzar, configurar i utilitzar les APIs que se'ns ofereixen des de Telecom Design, per tant, veiem el llistat d'APIs que tenen per saber quines utilitzarem.

5.3. Internet of Things API

Cada mòdul té una ID i una Key necessàries per registrar-lo i accedir al *Device Dashboard*, plataforma propietat de Telecom Design per accedir a les dades Sigfox. Aquests mòduls, a la vegada, es poden associar a un usuari, permetent al usuari gestionar tots les plaques registrades mitjançant el *Developer Dashboard*.

Tal com es pot veure a la documentació de Telecom Design sobre les seves APIs, tenim aquests apartats ben diferenciats. En el primer apartat, *Device API*, podem veure les característiques de les funcions relacionades amb els mòduls, mentre que en el segon, *Developer API*, veiem les funcions relacionades amb els usuaris registrats en aquesta web.

5.3.1. Device API

- *Authentication*: Cada mòdul té un seguit de lletres i números que el fan únic. Aquest identificador té el nom de *Token*, i és necessari per totes les funcions que veurem a continuació, per tant, la primera funció *Authentication* ens servirà per conèixer aquest *Token* mitjançant un JSON i així poder utilitzar les altres APIs.
- *Get PAC serial number*: Com el seu nom indica, el PAC és una altra cadena de caràcters que fa d'identificador, però no farà falta per la resta d'aplicacions.
- *Fetching messages history*: Fent ús d'aquesta API obtindrem un arxiu en format JSON com el de la Fig 5.3, en el que veurem els missatges enviats pel nostre mòdul. També podem configurar paràmetres com el nombre de missatges que volem llegir, o la data màxima que tindran els missatges mostrats. Aquests paràmetres estaran definits a 20 missatges i a data d'avui si no els canviem.
- *Fetching recents messages*: Variant de la API anterior que retorna només l'últim missatge enviat per la placa.
- *Clear messages*: Serveix per eliminar tots els missatges enviats des del nostre mòdul i no retorna res.

Com acabem de veure, moltes de les respostes que rebrem, estaran en format JSON. Aquest format té la funció de gestionar molta informació de manera fàcil i eficaç.

JSON (*JavaScript Object Notation*) és un format per l'intercanvi de dades, bàsicament JSON les descriu amb una sintaxis dedicada que s'utilitza per identificar i gestionar aquestes dades. JSON va ser creat com a l'alternativa a XML i va generar un gran nombre de seguidors gràcies al seu fàcil ús en *JavaScript*. Una de les millors avantatges que té l'ús d'aquest format, és que pot ser llegit per qualsevol llenguatge de programació, per tant, pot ser utilitzat per l'intercanvi d'informació entre diferents tecnologies.

```
Returns      Array of iot messages
              Sample JSON response

[
  {
    "contrib": {
      "temp": "LOW"
    },
    "ctxt": {
      "active": true,
      "alerts": 0,
      "alerts_ack": 0,
      "battery": "UNKNOWN",
      "firstseen": "Mar 25, 2013 2:35:36 PM",
      "id": "125010855",
      "index": 6,
      "lastseen": "Mar 25, 2013 2:35:32 PM",
      "level": "UNKNOWN",
      "lost": 0,
      "losts": 0,
      "msgs": 0,
      "network": "UNKNOWN",
      "serial": "1FA0/6",
      "status": "UNDEFINED",
      "tamper": "UNKNOWN",
      "temp": "LOW",
      "type": "OTHER",
      "watched": false
    },
    "extra": {},
    "lvl": "34.49",
    "station": "003B",
    "type": "event:temp_low",
    "when": 1364222132034
  }
]
```

Fig. 5.3. Exemple de missatge en format JSON.

5.3.2. Developer API

- *Authentication*: De la mateixa manera que els mòduls tenen un *Token* únic i necessari per les altres APIs, també cada usuari en té un. Aquest *Token* però, actualment ja no es pot consultar mitjançant aquesta aplicació, si no que es pot visualitzar simplement entrant al *Developer Dashboard*.
- *Create lot App*: Cada desenvolupador pot crear *Apps* pels seus mòduls utilitzant aquesta API i així realitzar funcions diferents a les que tenim aquí. El paràmetre necessari per aquesta comanda és un JSON amb el *Callback URL*, la descripció i el nom de l'*App*.
- *Delete lot App*: Per borrar qualsevol *App* creada anteriorment només fa falta cridar aquesta comanda amb la *ID* tècnica de l'*App*.
- *List lot Apps*: Amb aquesta API rebrem un JSON amb la llista d'*Apps* que hem creat i per tant tenim disponibles.
- *Register modules*: Introduint la *ID* i *Key* d'un mòdul el podem assignar al nostre usuari gràcies a aquesta API. També podrem fer-ho més intuïtivament des del *Developer Dashboard*.
- *Unregister module*: De la mateixa manera, també podrem alliberar un mòdul amb aquesta funció, però mentre el *Register modules* acabava retornant una llista amb els mòduls registrats, aquesta no retorna res.
- *List Registered modules*: Fent servir aquesta comanda rebrem un JSON amb la llista dels mòduls que tenim associats al nostre usuari.

5.4. Implementació en format PHP

PHP (acrònim de *Hypertext Preprocessor*) és un llenguatge de codi obert molt popular especialment adequat pel desenvolupament web que pot ser aplicat dins de codi HTML.

En comptes d'utilitzar moltes comandes per mostrar HTML, com passa amb C o amb Perl, les pàgines de PHP contenen HTML amb codi incrustat per crear *scripts* que facin tasques que no es poden fer amb HTML per si sol. L'avantatge més gran d'utilitzar PHP és la seva extrema simplicitat pel principiant, però sense deixar d'oferir moltes característiques avançades pels programadors professionals [10].

Com hem vist en l'apartat anterior, tenim dos grups d'APIs. Una futura aplicació per aquest projecte seria gestionar un gran número de mòduls i així fer més extens l'abast del projecte. D'aquesta manera seria molt útil fer ús de les Developer APIs per fer-ho tot més senzill.

En aquest cas, però, només disposem d'un mòdul i volem estudiar com configurar l'enviament i la rebuda de dades. Per aquest motiu, només ens serà necessari conèixer el nostre *Token* utilitzant la funció *Authentication* de l'apartat *Device API* i així poder cridar la *Fetching messages history*.

Una vegada cridada aquesta funció, haurem de gestionar el JSON rebut i triar-ne les dades que necessitem, que en el nostre cas seran tant el nivell de bateria com la posició GPS del nostre mòdul.

Per fer tot això, caldrà fer unes configuracions prèviament, per tant, anem a veure com es comportarà la nostra pàgina web pas per pas per entendre millor quins passos segueix.

Primerament, iniciem l'arxiu per poder escriure en llenguatge PHP. Tot seguit introduïm la *ID* i la *KEY* del nostre mòdul. En un futur, al tenir més d'un mòdul, aquestes dades seran entrades mitjançant un formulari previ o un inici de sessió. En aquest cas les escrivim al inici del codi donat que només disposem d'una placa.

```
<?php
// Credencials Mòdul
$ID=E532;
$KEY=A42FE840;
```

Per fer la crida de les APIs que acabem de veure, farem ús de la llibreria de funcions cURL, per tant, el primer que haurem de fer serà inicialitzar la crida mitjançant aquesta llibreria. Per fer-ho, és tan fàcil com guardar en una variable la funció inicialització.

```
// Inicialitzem el cURL
$curl = curl_init();
```

Una vegada ja la tenim inicialitzada, caldrà configurar-la mitjançant l'entrada d'uns paràmetres necessaris per preparar-la. Gracies al *curl_setopt_array*, podem entrar més d'un paràmetre alhora per estalviar espai i fer el codi més compacte. *CURLOPT_RETURNTRANSFER* ens retornarà el resultat de la transferència transformat a *string* si li entrem un 1 o un *True* i *CURLOPT_SLL_VERIFYPEER* no verificarà el peer del certificat si el configurem amb un *False*, i per tant no ens aturarà el procés de la web.

Fet això, haurem d'indicar l'adreça a la que s'ha de dirigir per cridar l'API que pertoca. En aquest cas, volem saber el *token* de la nostra placa per demanar després la llista de missatges d'aquesta. Com hem vist, per conèixer el *token* fan falta tant el *ID* del mòdul com la seva *KEY*, per tant, referenciem aquestes dades a la mateixa URL en el *CURLOPT_URL*

```
// El configurem i demanem el Token
curl_setopt_array($curl, array(
    CURLOPT_RETURNTRANSFER => 1,
    CURLOPT_SSL_VERIFYPEER => FALSE,
    CURLOPT_URL =>
"https://sensor.insgroup.fr/iot/devices/crc.json?sn=$ID&key=$KEY",
));
```

Com haurem d'utilitzar el *token* més endavant, el guardem a una variable utilitzant el *curl_exec*

```
// Guardem el token
$token = curl_exec($curl);
```

Per fer la crida del *Fetching messages history*, hauríem de configurar d'igual manera les funcions cURL, però ens serveix la configuració que hem fet anteriorment i per tant ens la estalviem.

M'agradaria remarcar que tot aquest procés ha estat molt complicat de confeccionar, degut a la recent existència d'aquestes tecnologies, però el que més temps ens ha portat sense dubtes ha sigut la manera en la que cridem la següent funció, ja que la documentació era ridículament escassa.

Per obtenir la llista de missatges de la nostra placa haurem de configurar l'encapçalament *CURLOPT_HTTPHEADER*, on indiquem que el missatge rebut serà en format JSON i on també indiquem quin serà el nostre *token*. Després escrivim la URL que pertoca, referenciant com abans, el *ID* i la *KEY* del mòdul.

```
// Tornem a configurar i demanem la llista de missatges
curl_setopt_array($curl, array(
    CURLOPT_HTTPHEADER => array(
        'Content-Type:application/json',
        "X-Snsr-Device-Key : $token",
    ),
    CURLOPT_URL =>
"https://sensor.insgroup.fr/iot/devices/msgs/history.json?sn=$ID&key=$KEY",
));
```

Ja tenim la llista dels missatges enviats pel nostre mòdul, per tant, aquí acaba el codi relacionat amb l'internet de les coses i per tant la part complicada d'aquest seguit de comandements. Abans de res, guardem aquesta llista i tanquem el cURL amb el *curl_close*.

```
// Guardem la llista de missatges i tanquem el cURL
$resp = curl_exec($curl);
curl_close($curl);
```

Una vegada tenim la llista guardada, el que volem és extreure'n les dades que necessitem, per tant, començarem obtenint la última posició correcte que hagi enviat la placa. Cada posició enviada per aquesta consta d'una latitud, una longitud i una altitud, però per mostrar la nostre posició en un mapa només fan falta les dos primeres i per això seran les que emmagatzemarem.

```
// Agafem la última posició correcte
$i=20;
$pos1=0;

while ($i>0):
$pos1 = stripos($resp, "latitude",$pos1)+11;
$pos2 = stripos($resp, ",", $pos1)-1;
$latitude=substr($resp, $pos1,$pos2-$pos1);

if ($latitude!=""):
$pos3 = $pos2 + 15;
$pos4 = stripos($resp, ",", $pos3)-1;
$longitude= substr($resp, $pos3, $pos4-$pos3);
$i=$i-30;
endif;

$i=$i-1;
$pos1=$pos1+1;
endwhile;
```

Pel que fa a la bateria, com ja hem estipulat anteriorment, enviarem els missatges de dos maneres diferents, en ASCII i en format digital pur. D'aquesta manera, reconeixem si l'últim estat de la bateria ha estat en ASCII per així guardar-lo en una variable, o per contra veurem que el format entrat és el digital i per tant hauréem de fer unes operacions prèvies per saber el tant per cent (SoC) de bateria real. Finalment, conclourem l'escrit PHP.

```
// Agafem l'últim valor del estat de la bateria
$pos5= stripos($resp, "asciiraw")+11;
$raw= substr($resp, $pos5, 6);
if ($raw===SoCASC):
$battery= substr($resp, $pos5+6,3);
if ($battery<100):
$battery= substr($resp, $pos5+7,2);
endif;
endif;
if ($raw===SoCDIG):
$x1= hexdec(substr($resp, $pos5-49,2));
$x2= hexdec(substr($resp, $pos5-47,2));
$k1= hexdec(substr($resp, $pos5-45,2));
$k2= hexdec(substr($resp, $pos5-43,2));
$battery= round(($x1*256+$x2)*100/($k1*256+$k2));
endif;
```

?>

Per tal que es vegi la longitud del codi de la nostra web, així com la part en la que configurem l'aspecte d'aquesta i utilitzem les funcions anteriors per actualitzar els *gadgets* que descobrirem en el següent apartat, mostraré tot el codi actual de la web ja dissenyada.

```
<?php
// Credencials Mòdul
$ID=E532;
$KEY=A42FE840;

// Inicialitzem el cURL
$curl = curl_init();

// El configurem i demanem el Token
curl_setopt_array($curl, array(
    CURLOPT_RETURNTRANSFER => 1,
    CURLOPT_SSL_VERIFYPEER => FALSE,
    CURLOPT_URL =>
"https://sensor.insgroup.fr/iot/devices/crc.json?sn=$ID&key=$KEY",
));

// Token E532: MDAwMEU1MzI6QTQyRkU4NDA=
// Token E4A3: MDAwMEU0QTM6Q0ZFM0E50TU=

// Guardem el token
$token = curl_exec($curl);

// Tornem a configurar i demanem la llista de missatges
curl_setopt_array($curl, array(
    CURLOPT_HTTPHEADER => array(
        'Content-Type:application/json',
        "X-Snsr-Device-Key : $token",
    ),
    CURLOPT_URL =>
"https://sensor.insgroup.fr/iot/devices/msgs/history.json?sn=$ID&key=$KEY",
));

// Guardem la llista de missatges i tanquem el cURL
$resp = curl_exec($curl);
curl_close($curl);

// Agafem la última posició correcte
$i=20;
$pos1=0;

while ($i>0):
    $pos1 = strpos($resp, "latitude",$pos1)+11;
    $pos2 = strpos($resp, ",", $pos1)-1;
    $latitude=substr($resp, $pos1,$pos2-$pos1);

    if ($latitude=="0"):
        $pos3 = $pos2 + 15;
        $pos4 = strpos($resp, ",", $pos3)-1;
        $longitude= substr($resp, $pos3, $pos4-$pos3);
        $i=$i-30;
    endif;
```

```

$i=$i-1;
$pos1=$pos1+1;
endwhile;

// Agafem l'últim valor del estat de la bateria
$pos5= strpos($resp, "asciiraw")+11;
$raw= substr($resp, $pos5, 6);
if ($raw===SoCASC):
$battery= substr($resp, $pos5+6,3);
if ($battery<100):
$battery= substr($resp, $pos5+7,2);
endif;
endif;
if ($raw===SoCDIG):
$x1= hexdec(substr($resp, $pos5-49,2));
$x2= hexdec(substr($resp, $pos5-47,2));
$k1= hexdec(substr($resp, $pos5-45,2));
$k2= hexdec(substr($resp, $pos5-43,2));
$battery= round(($x1*256+$x2)*100/($k1*256+$k2));
endif;
?>

<!DOCTYPE html>
<html>
<head>

<!--Battery Level-->

<script src="js/AquaGauge.js" type="text/javascript"></script>
<script language="javascript" type="text/javascript">
function showGauge() {
    var aGauge = new AquaGauge('gauge');
    aGauge.props.minValue = 0;
    aGauge.props.maxValue = 100;
    aGauge.props.dialTitle = "Battery[%]";
    aGauge.props.dialSubTitle = "By Edgar Tamayo";
    aGauge.props.rimWidth = 10;
    aGauge.props.rangeSegments[0].start = 0;
    aGauge.props.rangeSegments[0].end = 20;
    aGauge.props.rangeSegments[0].color = "red";
    aGauge.props.rangeSegments[1].start = 20;
    aGauge.props.rangeSegments[1].end = 50;
    aGauge.props.rangeSegments[1].color = "yellow";
    aGauge.props.rangeSegments[2].start = 50;
    aGauge.props.rangeSegments[2].end = 100;
    aGauge.props.rangeSegments[2].color = "greenyellow";
    aGauge.refresh(<?php echo $battery; ?>);
}
</script>

<!--Map-->

<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />

```

```

<style type="text/css">
  html { height: 100% }
  body { height: 100%; margin: 0; padding: 0 }
  #mapa_div { height: 100% }
</style>
<script type="text/javascript"
  src="http://maps.googleapis.com/maps/api/js">
</script>
<script type="text/javascript">
  var marker = null;

  function funcionClick() {
    if (marker.getAnimation() != null) {
      marker.setAnimation(null);
    } else {
      marker.setAnimation(google.maps.Animation.BOUNCE);
    }
  }

  function inicializar_mapa() {
    var mapOptions = {
      center: new google.maps.LatLng(<?php echo $latitude;?>, <?php echo
$longitude;?>),
      zoom: 15,
      mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    var map = new google.maps.Map(document.getElementById("mapa_div"),
      mapOptions);

    var pos = new google.maps.LatLng(<?php echo $latitude;?>, <?php echo
$longitude;?>);

    marker = new google.maps.Marker({
      position: pos,
      map: map,
      title:"Here is your Bultaco",
      animation: google.maps.Animation.DROP
    });

    marker.setIcon('motorbike.png');
    google.maps.event.addListener(marker, 'click', funcionClick);
  }
</script>
</head>

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="http://www.w3schools.com/lib/w3.css">

<body onload="showGauge();inicializar_mapa()">

<div class="w3-container w3-orange">
  <h1>Bultaco Rapidan</h1>
  <p>Save the planet on two wheels!</p>

```



```

</div>

<div class="w3-row-padding">

<div class="w3-third">
  <h2>Battery</h2>
  <canvas id="gauge" width="300" height="300">Browser not supported.</canvas>
  <p>Autonomy: <?php echo $battery*2; ?> km
    <br>
    <a href="http://www.electromaps.com/"> Where to recharge?</a></p>
</div>

  <!--Lasts Messages-->

<div class="w3-third">
  <h2>Lasts Messages</h2>
  <p style="border: 1px solid;padding: 10px">
    Battery: <?php echo $battery;?><br>
    Latitude: <?php echo $latitude;?><br>
    Longitude: <?php echo $longitude;?></p>
</div>
  <br><br><br><br><br><br><br><br><br>

<div class="w3-third">
  <h2>Actual position</h2>
  <div id="mapa_div" style="width:100%;height:200px;"></div>
</div>

</div>

</body>
</html>

```

6. Pàgina web

La idea principal d'aquet projecte és la de consultar dades d'una motocicleta elèctrica a temps real mitjançant el "Internet of things".

Per poder mostrar aquestes dades, primerament es va pensar en dissenyar una app Android i així poder rebre les dades a qualsevol terminal mòbil Android. Després d'estudiar aquesta possibilitat però, es va arribar a la conclusió que podria endarrerir l'entrega del projecte degut a els poques nocions sobre programació d'Android de que disposava, i que per tant, seria més eficaç dissenyar una pàgina web amb HTML per a la primera etapa del projecte.

Tot i això, pel disseny d'aquesta pàgina web també van fer falta moltes hores de llegir documentació sobre la construcció de webs amb llenguatge HTML5 actual, però primer es va pensar en les dades que s'hi haurien de mostrar i en la seva estructuració.

Per l'usuari d'una motocicleta elèctrica, així com per l'empresa subministradora de motocicletes, es va pensar que la informació prioritària a mostrar per pantalla hauria de ser el nivell de bateria i la ubicació del vehicle. Per altra banda, per fer proves d'entrada de dades faria falta un qüestionari i una quadre de text on es mostressin els últims missatges entregats per la placa implementada en la motocicleta. Per últim, s'hauria de cuidar l'estètica i separar per apartats les diferents funcions.

6.1. Nivell de bateria

Per l'usuari habitual d'una motocicleta elèctrica, el més important és conèixer l'estat de la bateria o SoC (State of Charge). Per aquesta raó, aquesta informació s'ha de mostrar de forma molt visible mitjançant un gadgetgadget relativament gran, amb variacions de formes o colors al sobrepassar uns llindars de bateria i sobretot mostrar clarament el percentatge de bateria o els kilòmetres d'autonomia de que disposa la motocicleta.

Es van estar provant diferents gadgetgadgets per HTML5, com per exemple el que s'observa a la Fig. 6.1 [11], però la dificultat de fer-lo servir per poder-lo implementar dins la web va fer que es busquessin altres exemples més senzills de personalitzar i amb una documentació més clara.

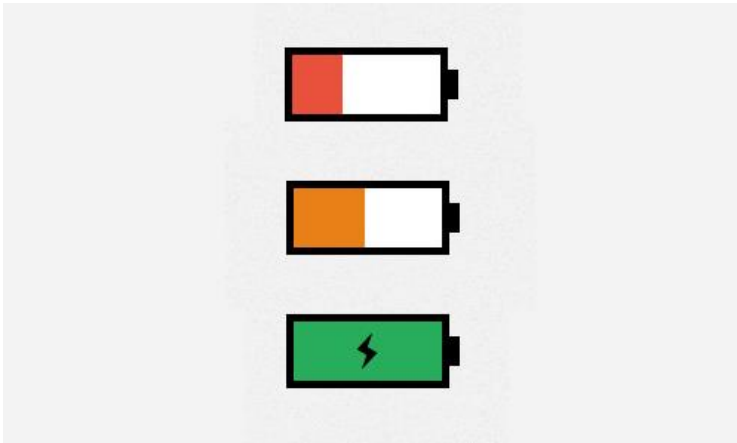


Fig. 6.3. Gadget per la bateria de www.hongkiat.com

Seguidament, es va trobar a la pàgina web de Code Project [12] el codi necessari per fer funcionar un gadget anomenat *Agua Gauge*. Aquest era molt més fàcil de personalitzar donat que permetia canviar els diferents llindars pels nivells de bateria baixa, mitja, o alta, els colors de fons, els nivells màxims i mínims de bateria i la mida del mateix gadget, com s'observa a la Fig. 6.2.

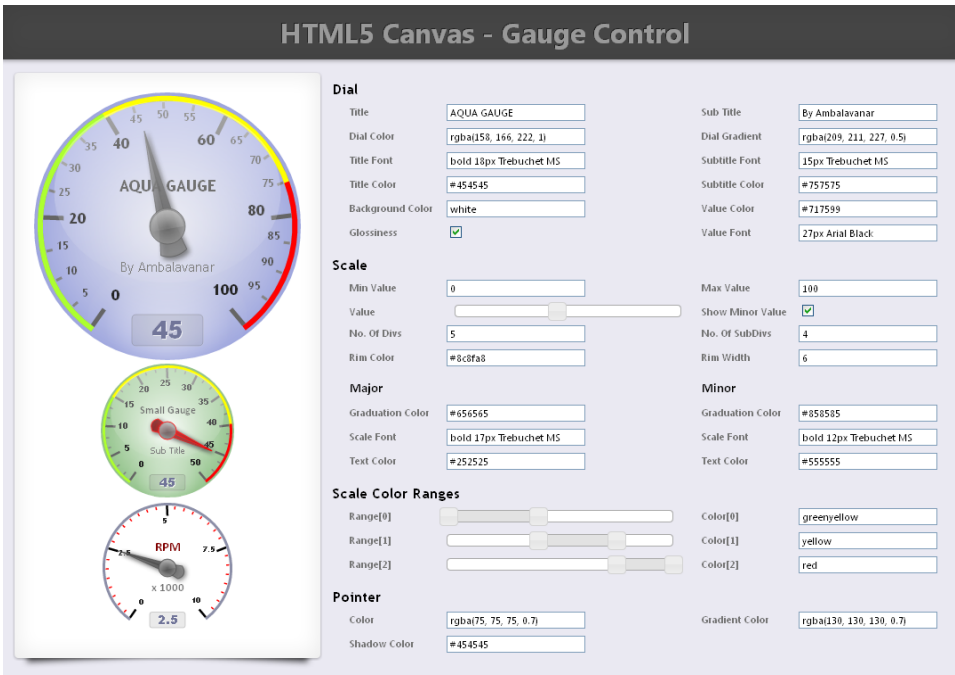


Fig. 6.2. Gadget per la bateria de www.codeproject.com

```
<!--Battery Level-->
```

```
<script src="js/AquaGauge.js" type="text/javascript"></script>
<script language="javascript" type="text/javascript">
function showGauge() {
    var aGauge = new AquaGauge('gauge');
    aGauge.props.minValue = 0;
    aGauge.props.maxValue = 100;
    aGauge.props.dialTitle = "Battery[%]";
    aGauge.props.dialSubTitle = "By Edgar Tamayo";
    aGauge.props.rimWidth = 10;
    aGauge.props.rangeSegments[0].start = 0;
    aGauge.props.rangeSegments[0].end = 20;
    aGauge.props.rangeSegments[0].color = "red";
    aGauge.props.rangeSegments[1].start = 20;
    aGauge.props.rangeSegments[1].end = 50;
    aGauge.props.rangeSegments[1].color = "yellow";
    aGauge.props.rangeSegments[2].start = 50;
    aGauge.props.rangeSegments[2].end = 100;
    aGauge.props.rangeSegments[2].color = "greenyellow";
    aGauge.refresh(<?php echo $_POST["batt"]; ?>);
}
</script>
```

També es va creure convenient mostrar l'autonomia de la motocicleta per facilitar l'organització de viatges a l'usuari. Per fer això es va consultar l'autonomia amb un 100% de bateria de una motocicleta elèctrica qualsevol, en aquest cas una Bultaco Rapidan i mitjançant el llenguatge PHP i el percentatge de la bateria actual de la motocicleta es calcula l'autonomia aproximada.

Battery



Autonomy: 152 km
[Where to recharge?](#)

Per acabar, donades les poques infraestructures per recarregar vehicles elèctrics, es va veure convenient indicar un enllaç a una pàgina web [13] per consultar on són aquests pàrquings elèctrics i de quin tipus són.

Fig. 6.3. Gadget preparat per implementar

6.2. Localització GPS

En cas de no saber on està la motocicleta, per haver-la deixat a algú, no saber on està aparcada o fins i tot en cas de robatori, la pàgina web ha de mostrar la ubicació d'aquesta.

Gràcies a la placa TD1204, podem saber en tot moment la latitud, longitud i altitud del vehicle, per tant, una bona implementació d'aquest servei seria mostrar un mapa i ubicar la motocicleta en aquest gràcies a les coordenades rebudes des de Sigfox.

Per aquesta aplicació es va decidir fer ús de les APIs que ofereix Google Maps. Mitjançant exemples i tutorials, es va aprendre a personalitzar un *script* per mostrar per pantalla un mapa de la mida apropiada, centrat al punt geogràfic desitjat i amb un zoom determinat [14].

Per tal que fos estèticament més atractiu es va canviar el marcador de posició original pel d'una motocicleta, es va dissenyar un text per quan el cursor es troba sobre el marcador i es van personalitzar les animacions d'aquest per tal d'emfatitzar la posició del vehicle dins del mapa.

Actual position



Fig. 6.4. Gadget per la localització de Google Maps

```

<!--Map-->

<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<style type="text/css">
  html { height: 100% }
  body { height: 100%; margin: 0; padding: 0 }
  #mapa_div { height: 100% }
</style>
<script type="text/javascript"
  src="http://maps.googleapis.com/maps/api/js">
</script>
<script type="text/javascript">
  var marker = null;

  function funcionClick() {
    if (marker.getAnimation() != null) {
      marker.setAnimation(null);
    } else {
      marker.setAnimation(google.maps.Animation.BOUNCE);
    }
  }

  function inicializar_mapa() {
    var mapOptions = {
      center: new google.maps.LatLng(<?php echo $_POST["lat"]; ?>, <?php
echo $_POST["long"]; ?>),
      zoom: <?php echo $_POST["zoom"]; ?>,
      mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    var map = new google.maps.Map(document.getElementById("mapa_div"),
      mapOptions);

    var pos = new google.maps.LatLng(<?php echo $_POST["lat"]; ?>, <?php
echo $_POST["long"]; ?>);

    marker = new google.maps.Marker({
      position: pos,
      map: map,
      title:"Here is your Bultaco",
      animation: google.maps.Animation.DROP
    });

    marker.setIcon('motorbike.png');
    google.maps.event.addListener(marker, 'click', funcionClick);

  }
</script>

```

6.3. Quadre de text i qüestionari

De cara a comprovar el funcionament de les diferents funcions de la web per poder conèixer i arreglar els errors que es van originant, és necessari anar fent petites proves.

Per una banda, es dissenya un quadre de text per poder llegir els últims missatges que la web rep de la TD1204, tals com les coordenades, el nivell de bateria i d'altres. D'aquesta manera podem conèixer el format en que arriben aquetes dades per poder-les utilitzar per actualitzar les altres funcions.

D'altra banda, mentre no rebem informació del servidor de Sigfox, podem entrar les dades del quadre de text, el gadget de la bateria i el mapa de Google mitjançant un qüestionari. Per tant, utilitzant el mètode POST amb el llenguatge PHP es dissenya una entrada de dades per tal d'imitar la TD1204 amb un botó "Submit" que redirigeix a la mateixa web però amb dades actualitzades.

Battery:	67
Message 1:	Bateria: 67
Message 2:	Latitud: 41.385064
Message 3:	Longitud: 2.173403
Latitud:	41.385064
Longitud:	2.173403
Zoom:	2

Submit

Fig. 6.5. Qüestionari per entrar dades

<!--Questionnaire-->

```
<div class="w3-third w3-container">
  <form action="index.php" method="post">
    <br><br><br>
    Battery: <input type="text" name="batt"><br>
    Message 1: <input type="text" name="msg1"><br>
    Message 2: <input type="text" name="msg2"><br>
    Message 3: <input type="text" name="msg3"><br>
    Latitud: <input type="text" name="lat"><br>
    Longitud: <input type="text" name="long"><br>
    Zoom: <input type="text" name="zoom"><br>
    <br>
  <input type="submit" value="Submit">
  <br>
</form>
</div>
```

Lasts Messages

Bateria: 67
 Latitud: 41.385064
 Longitud: 2.173403

Fig. 6.6. Últims missatges rebuts per la web

```
<!--Lasts Messages-->

<div class="w3-third w3-container">
  <h2>Lasts Messages</h2>
  <p style="border: 1px solid; padding: 10px">
    <?php echo $_POST["msg1"]; ?><br>
    <?php echo $_POST["msg2"]; ?><br>
    <?php echo $_POST["msg3"]; ?></p>
</div>
```


6.4. Format i distribució

Un cop definides totes les funcions, calia estudiar la seva estructuració i procurar de millorar l'aspecte gràfic per aconseguir que fos més visual i ergonòmic. Amb les nocions de que disposava, només aconseguia encadenar els gadgets i canviar el color de fons. Per tant es va decidir mirar plantilles i tutorials.

Des de la web www.w3schools.com, ensenyen a dissenyar pas per pas una pàgina agradable i eficient. Es va aprendre per tant, a estructurar la web en blocs diferenciats, que canviessin de mida o de disposició per adaptar-se a la pantalla, el qual és molt recomanable per gaudir d'una bona presència en les pantalles de "tablets" o "smartphones". Tot seguit es va fer ús d'una plantilla per fer-ho més ràpid i intuïtiu i es va implementar i provar amb el programa Microsoft WebMatrix.

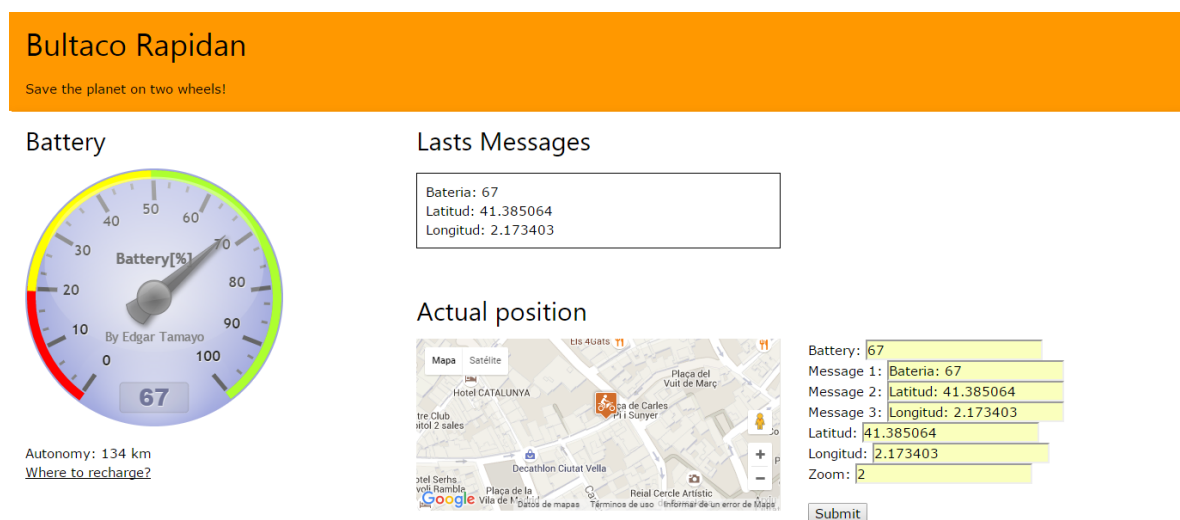


Fig. 6.7. Previsualització de la web preparada per fer proves

Conclusions

Després de mesos de recerca d'informació i de treball amb aquesta tecnologia que és l'Internet de les Coses, puc concloure amb total seguretat, que estem davant d'una gran corrent de desenvolupament tecnològic que donarà lloc a in comptables innovacions.

En un futur, tots els objectes quotidians que comprem estaran connectats a internet per comunicar-se entre ells i amb nosaltres. Gaudirem de tota la informació proporcionada pels sensors de que disposin els nostres electrodomèstics, vehicles o aparells en general, en un sol terminal, com pot ser el nostre *smartphone*.

Vaig escollir aquest projecte dels professors Manuel Moreno i Emili Hernandez perquè em semblava una tecnologia interessantíssima. I no sols no m'ha decebut, si no que he après a utilitzar totes les eines informàtiques necessàries per poder dur a terme un projecte com aquest, i per tant, per realitzar qualsevol projecte d'aquest àmbit que pugui sorgir en un futur.

No obstant, la mateixa novetat d'aquesta tecnologia que fa que tinguem tantes possibilitats per innovar, també és la culpable que sigui molt complicat cercar informació relacionada amb aquest tema. Les pròpies webs responsables dels mòduls que hem fet servir es veuen massa noves, amb poca documentació per a programadors no professionals i no gaire ben organitzades.

També cal a dir que han hagut problemes amb la cobertura, tals com no poder enviar cap tipus de missatge des d'una ubicació concreta o només poder calcular la posició GPS en espais a cel obert.

Tret dels problemes esmentats, aquest projecte ha acabat sent el que volia ser, una visió propera de la tecnologia estudiada, una aplicació pràctica relacionada amb el món dels vehicles elèctrics i un ajut pels pròxims desenvolupadors que pateixin la manca de documentació, que trobaran en aquest treball unes pautes a seguir per estalviar molt temps de recerca.

Però no només això, ja que el que més m'entusiasma d'haver realitzat aquest projecte, ha estat que ara em sento capaç de poder desenvolupar totes les idees que em puguin passar pel cap, per a acabar sent projectes reals.

Agraïments

Per concloure aquest treball, m'agradaria agrair primerament al meu tutor i director en aquest projecte, Manuel Moreno, que m'ha aconsellat i ajudat en tots els moments en que no sabia per on continuar el meu treball, i que per tant no hagués estat possible sense ell.

També agraeixo a l'Emili Hernández, codirector d'aquest projecte, que m'hagi animat en cada reunió mostrant-me el gran potencial de l'Internet de les Coses i l'interès que mostren les empreses per treballs relacionats amb aquesta matèria.

A l'empresa Telecom Design que ens han proporcionat dos mòduls TD1204 per a desenvolupadors essencials i necessaris per poder realitzar aquest projecte, així com ens ha intentat ajudar en tot el possible per resoldre tots els problemes que se'ns anaven presentant.

Per últim i no menys important, agrair a Israel Terán Lozano i a Albert Torres Pons, per tot el seu treball en aquest camp de l'Internet de les Coses i per tota la ajuda que m'han proporcionat quan més la necessitava.

Bibliografia

- [1] TFM: *Internet de les coses: una primera aproximació utilitzant la tecnologia SIGFOX*, d'Israel Terán Lozano, 2015.
- [2] TFG: *Internet de les coses aplicat a la millora del servei de Bicing de Barcelona*, d'Albert Torres Pons, 2015.
- [3] DGT. Parque de vehiculos, 2015. [<http://www.dgt.es/es/explora/en-cifras/parque-de-vehiculos.shtml>], 26 de maig de 2015]
- [4] 20 MINUTOS. Evolución del precio de los carburantes, 2013. [<http://www.20minutos.es/graficos/evolucion-del-precio-de-los-carburantes-7/0/>], 10 de juny de 2015]
- [5] 20 MINUTOS. ¿Cuánto pagaré de electricidad si compro un coche eléctrico?, 2014. [<http://blogs.20minutos.es/coches-electricos-hibridos/2014/01/13/cuanto-pagare-de-electricidad-si-compro-un-coche-electrico/>], 11 de juny de 2015]
- [6] TELECOM DESIGN. TD Next Sigfox Modules, 2015. [<http://rfmodules.td-next.com/>], 15 de juny de 2015]
- [7] TELECOM DESIGN. Device Dashboard, 2015. [<https://developers.insgroup.fr/dashboards/device.html>], 24 de maig de 2015]
- [8] TELECOM DESIGN. Device API, 2015. [<https://developers.insgroup.fr/iot/device.html>], 20 de juny de 2015]
- [9] PHP.NET. Biblioteca URL Cliente, 2013 [<http://php.net/manual/es/book.curl.php>], 3 de juliol 2015]
- [10] PHP.NET. PHP 7.0.0, 2015 [<http://php.net/>], 27 de juny de 2015]
- [11] HONGKIAT. How to retrieve and display battery status with HTML5, 2013. [<http://www.hongkiat.com/blog/html5-battery-status/>], 17 de juny de 2015]
- [12] CODE PROJECT. Gauge control using HTML5 Canvas, 2011. [<http://www.codeproject.com/Articles/304874/HTML-Canvas-Aqua-Gauge>], 19 de juny de 2015]
- [13] ELECTROMAPS. Todo sobre el vehículo eléctrico y puntos de recarga,



2015. [www.electromaps.com, 1 de juliol de 2015]

- [14] GOOGLE DEVELOPERS. Google Maps Javascript API, 2015. [<https://developers.google.com/maps/documentation/javascript/tutorial>, 7 de juliol de 2015]